

533,995

(12) DE L'UNION INTERNATIONALE PUBLIÉE EN VERTU DU TRAITÉ DE COOPÉRATION  
EN MATIÈRE DE BREVETS (PCT)

(19) Organisation Mondiale de la Propriété  
Intellectuelle  
Bureau international



(43) Date de la publication internationale  
27 mai 2004 (27.05.2004)

PCT

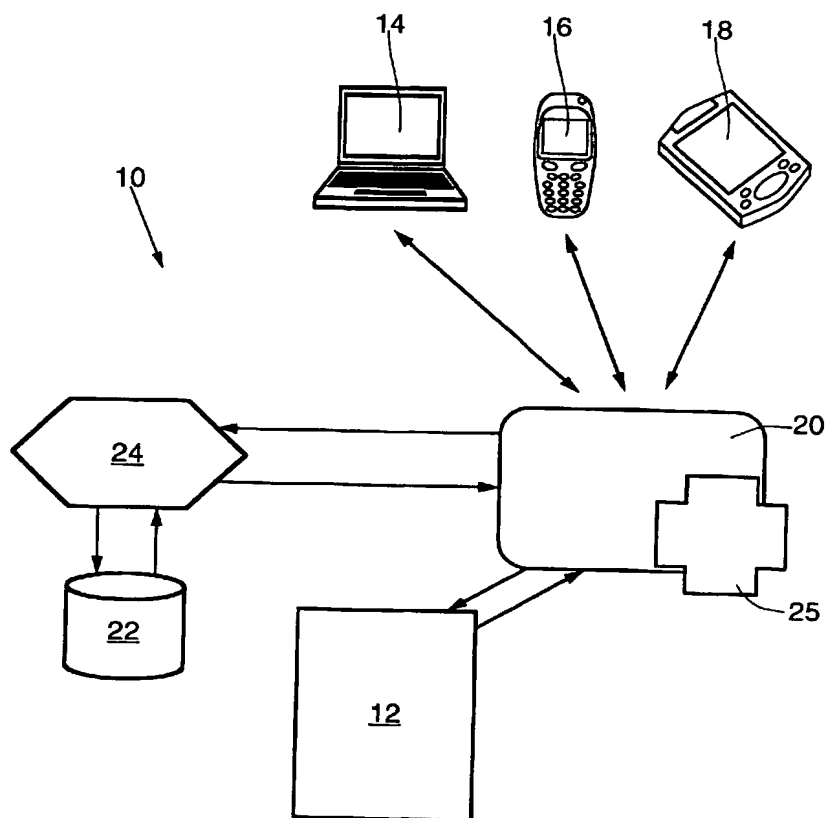
(10) Numéro de publication internationale  
WO 2004/044789 A2

- (51) Classification internationale des brevets<sup>7</sup> : G06F 17/50 (71) Déposant (pour tous les États désignés sauf US) : FRANCE TELECOM [FR/FR]; 6, place d'Alleray, F-75015 Paris (FR).
- (21) Numéro de la demande internationale : PCT/FR2003/003305 (72) Inventeurs; et (75) Inventeurs/Déposants (pour US seulement) : TRAN XUAN, Fabrice [FR/FR]; 16, rue Etienne de la Boétie, F-38320 Eybens (FR). SOLER, Karine [FR/FR]; Le Village, F-38210 Montaud (FR). SIMON, Christophe [FR/FR]; 230, chemin de la Pelado, F-84270 Vedene (FR).
- (22) Date de dépôt international : 5 novembre 2003 (05.11.2003)
- (25) Langue de dépôt : français
- (26) Langue de publication : français
- (30) Données relatives à la priorité : 02/13872 6 novembre 2002 (06.11.2002) FR (74) Mandataire : BUREAU D.A. CASALONGA-JOSSE; 8, avenue Percier, F-75008 Paris (FR).
- (81) États désignés (national) : AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ,

[Suite sur la page suivante]

(54) Title: DYNAMIC IMAGE PRODUCTION METHOD AND SYSTEM

(54) Titre : PROCEDE ET SYSTEME D'ELABORATION DYNAMIQUE D'IMAGES



(57) Abstract: The invention relates to a system which is used for the dynamic production of images and the transmission of said images to remote terminals (14, 16, 18). The inventive system consists of a server comprising an image database (12) and means (12, 20, 25) which are used to process images intended for the terminals and which can convert said images into respective formats that are compatible with the remote terminals. The aforementioned image database is a database of vector images. The above-mentioned processing means comprise a set of tags which are integrated into the vector images and which cause the images to be manipulated in order to make same compatible with the remote terminals and means of generating images which can be read by the terminals from the manipulated images.

(57) Abrégé : Ce système d'élaboration dynamique d'images d'images et de transmission de ces images vers des terminaux distants (14, 16, 18) comprend un serveur comportant une base d'images (12) et des moyens de traitement (12, 20, 25) des images destinées aux terminaux adaptés pour convertir les images en formats

respectifs compatibles avec les terminaux distants. La base d'images

[Suite sur la page suivante]

WO 2004/044789 A2



DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

TR), brevet OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

**Publiée :**

— *sans rapport de recherche internationale, sera republiée dès réception de ce rapport*

- (84) **États désignés (régional) :** brevet ARIPO (BW, GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), brevet eurasién (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), brevet européen (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK,

*En ce qui concerne les codes à deux lettres et autres abréviations, se référer aux "Notes explicatives relatives aux codes et abréviations" figurant au début de chaque numéro ordinaire de la Gazette du PCT.*

## Procédé et système d'élaboration dynamique d'images

L'invention concerne un système et un procédé d'élaboration dynamique d'images.

5 Plus particulièrement, l'invention se rapporte à un système permettant d'élaborer des images à partir d'un fichier source et de télécharger ces images vers des terminaux distants pour leur affichage.

10 Une application particulièrement intéressante d'un tel procédé et d'un tel système concerne l'élaboration d'images à partir de fichiers sources extraits de sites de réseaux informatiques tels que le réseau Internet et la télétransmission de ces images vers des terminaux distants pour être consultés par des utilisateurs finaux.

15 Comme on le sait, la grande diversité des terminaux pouvant être utilisés pour la récupération de telles images ainsi que l'étendue géographique de tels réseaux informatiques, impliquent des contraintes majeures quant aux images finales à afficher sur les terminaux.

20 Il est en effet nécessaire que l'image finale tienne compte, d'une part, des contraintes des terminaux distants, relatives notamment à la taille de l'écran, sa résolution, le nombre de couleurs disponibles et, d'autre part, de la langue dans laquelle l'image doit être affichée sur les terminaux de manière à être compréhensibles par l'utilisateur final.

25 En outre, des contraintes d'autres natures, liées par exemple à l'identité du ou des destinataires, peuvent également compliquer l'affichage des images. Ainsi, par exemple, il arrive parfois que des images soient toutes personnalisées de la même manière lorsqu'elles sont destinées à des utilisateurs d'un même groupe. Tel est en particulier le cas d'images destinées à être diffusées au sein de filiales d'une même société-mère pour lesquelles un même formalisme doit  
30 être utilisé.

Pour atteindre ces objectifs, il a été proposé d'utiliser des logiciels de conception graphique destinés à manipuler une image source pour l'exporter dans différents formats graphiques compatibles avec les terminaux distants. Comme on le conçoit, une telle technique

est relativement peu satisfaisante, dans la mesure où elle nécessite une élaboration complète des images à afficher à partir de l'image source. Elle ne tient en outre pas compte des problèmes de traduction d'éléments textuels.

5 D'autres types de techniques font appel à une génération dynamique d'images à partir de données reçues en entrée, en utilisant des programmes de traitement d'images spécifiques pour chaque type de terminal ou d'application. Une telle technique est néanmoins relativement coûteuse, dans la mesure où elle nécessite de prévoir un  
10 programme de traitement pour chaque image à générer.

Au vu de ce qui précède, le but de l'invention est de palier les inconvénients de l'état de la technique et de fournir un procédé et un système d'élaboration dynamique d'images et de transmission de ces images vers des terminaux distants permettant l'élaboration d'images  
15 selon des formats correspondants aux contraintes de terminaux qui leur sont destinés et des utilisateurs finaux, et ce de manière automatique, tout en ne nécessitant que des moyens matériels et logiciels relativement simples et pouvant en outre être utilisés de manière universelle, c'est-à-dire pour tous les types de terminaux pour  
20 satisfaire toutes les contraintes des utilisateurs.

Ainsi, selon l'invention, il est proposé un système d'élaboration dynamique d'images et de transmission de ces images vers des terminaux distants, comprenant un serveur associé à une base d'images et des moyens de traitement des images destinées aux terminaux  
25 adaptés pour convertir les images en format respectif compatible avec les terminaux distants.

La base d'images est une base d'images vectorielles. En outre, les moyens de traitement comportent un ensemble de balises insérées aux images de la base d'images et adaptées pour provoquer une  
30 manipulation des images pour les rendre compatibles avec les terminaux distants et des moyens pour générer des images lisibles par les terminaux à partir des images manipulées.

Pour l'élaboration d'une image pouvant être lue, par tout type de terminal, on insère ainsi des balises dans un fichier correspondant à

une image source pour élaborer une image « universelle », ces balises étant destinées à être reconnues par les moyens de traitement pour activer des programmes correspondants hébergés dans le serveur pour manipuler l'image source et ce, pour chaque type de terminal.

5           On conçoit dès lors qu'un seul programme de traitement d'images, qui fait appel à une pluralité de programmes de manipulation d'images appelés après détection des balises, n'est nécessaire pour l'élaboration de l'image finale.

10           Selon une autre caractéristique de l'invention, les images stockées dans la base d'images sont des images structurées selon une grammaire XML. Le format SVG constitue un format approprié pour structurer les données d'image dans la base d'images.

15           Selon une autre caractéristique du système selon l'invention, les moyens de traitement comportent en outre un ensemble de programmes de traitement d'images activés sélectivement par lesdites balises pour modifier en conséquence une image extraite de la base d'images.

20           Ces moyens de traitement comportent en outre des moyens d'élaboration d'une feuille de style adaptée pour insérer dans les images des informations de présentation.

          Selon un exemple de réalisation, ces feuilles de style constituent des moyens de transformation de type XSLT.

25           Les moyens d'élaboration de la feuille de style comportent par exemple, des moyens pour élaborer la feuille de style en fonction d'informations transmises à partir des terminaux distants.

          Selon un exemple de mise en œuvre, les moyens pour générer les images lisibles sont hébergées dans le serveur.

          En variante, si les terminaux le permettent, les moyens pour générer les images lisibles sont hébergées dans les terminaux.

30           Selon une autre caractéristique de l'invention, le serveur comporte en outre une mémoire cache pour le stockage des images élaborées par les moyens de traitement.

          On notera par ailleurs que lesdits programmes de traitement constituent des moyens pour provoquer une insertion de codes

d'instruction dans l'image extraite de la base de données en fonction d'au moins un argument des balises.

De préférence, les arguments des balises sont véhiculés par des requêtes transmises par les terminaux distants.

5 Selon l'invention, il est également proposé un procédé d'élaboration dynamique d'images et de transmission de ces images vers des terminaux distants. Ce procédé est caractérisé en ce qu'il comporte les étapes consistant à :

- 10 - stocker des images vectorielles dans une base d'images gérée par un serveur, les images comprenant chacune des balises adaptées pour provoquer une manipulation des images pour les rendre compatibles avec des terminaux distants pour l'affichage des images ;
- modifier les images en fonction des balises insérées ; et
- 15 - générer des images lisibles par les terminaux à partir des images dans lesquelles sont insérer les balises.

Au cours de l'étape de modification des images, le programme de traitement provoque l'élaboration d'un code de programmation et l'enrichissement dynamique d'un fichier d'image par insertion dudit code en remplacement des balises.

20 Selon un mode de mise en œuvre, les images lisibles par les terminaux sont élaborés par lesdits terminaux.

En variante, les images lisibles par les terminaux sont élaborés par le serveur.

25 D'autres but, caractéristiques et avantages de l'invention apparaîtront à la lecture de la description suivante, donnée uniquement à titre d'exemple non limitatif, et faite en référence aux dessins annexés sur lesquels :

- la figure 1 est une vue schématique d'un système d'élaboration et de transmission d'images conforme à l'invention ; et
- 30 - la figure 2 est un organigramme illustrant les principales phases d'un procédé mis en œuvre au moyen du système de la figure 1.

Sur la figure 1, on a représenté la structure générale d'un système d'élaboration et de transmission dynamique d'images conforme à l'invention, désigné par la référence numérique générale 10.

Il est destiné à extraire des images d'une base d'images 12 en réponse à des requêtes formulées par des terminaux distants tels que 14, 16 et 18, et à traiter ces images de manière à les rendre compatibles avec les contraintes de chaque terminal, en particulier en termes de résolution, de nombre de couleurs disponibles, de taille d'écran, tout en conservant la qualité d'une image initiale, mais également en tenant compte de contraintes d'autres natures, telles que des contraintes linguistiques, en procédant à une traduction dynamique d'éléments textuels contenus dans une image de manière à les rendre compréhensibles par l'utilisateur distant, et des contraintes liées à des formalismes prédéterminés imposés par un utilisateur émetteur d'une requête.

Comme on le voit sur la figure 1, les terminaux distants susceptibles à priori de recevoir les images peuvent être constitués par des terminaux informatiques de différentes natures. Ils peuvent en effet être constitués par des micro-ordinateurs, des téléphones mobiles cellulaires, des assistants personnels, ...

Chacun de ces équipements présente des caractéristiques et des capacités qui engendrent des contraintes particulières pour l'affichage d'images.

Ainsi, en réponse à des requêtes formulées par un utilisateur, le système 10 provoque une manipulation dynamique des images extraites de la base d'images 12 de manière à satisfaire l'ensemble des contraintes liées au terminal et à l'utilisateur concerné.

A cet effet, le système 10 comporte essentiellement un serveur, constitué par exemple par un serveur hébergeant un site Internet.

Comme on le voit sur la figure 1, le serveur comporte, outre la base d'images 12, un étage de traitement 20 assurant une récupération des images chargées dans la base d'images 12 pour les transformer en images se présentant selon un format les rendant susceptibles d'être affichées sur les terminaux 14, 16 et 18.

Cet étage de traitement est raccordé à une mémoire cache 22, par l'intermédiaire d'un étage de gestion 24 pour le stockage des images élaborées par l'étage de traitement 20.

En ce qui concerne la base d'images 12, celle-ci regroupe l'ensemble des images susceptibles d'être interrogées par les utilisateurs. Ces images sont chargées dans des fichiers structurés selon une grammaire XML (« eXtensible Markup Language », en anglais).

Plus particulièrement, les fichiers stockés dans la base d'image 12 sont structurés selon le format connu sous l'appellation SVG ("Scalable Vector Graphics", en anglais), c'est-à-dire sous la forme d'images vectorielles. Un tel format permet le stockage des images sous une forme plus compacte qu'une description pixels par pixels, telle qu'elle serait faite selon des formats GIF, JPEG, PNG,...

En outre, un tel format de stockage d'images rend possible la description d'une image en termes d'objets et autorise ainsi une manipulation des images en travaillant sur les éléments qui la composent, contrairement aux formats d'images selon lesquels les images sont décrites pixel par pixel.

Les fichiers stockés dans la base d'image 12 sont élaborés à partir de fichiers source d'origine regroupant, sous une forme vectorielle, c'est-à-dire selon le format SVG, l'ensemble des images disponibles.

Comme cela est connu en soit, un fichier SVG est une application du langage XML. Elle se présente sous la forme d'un fichier-texte dont l'édition est possible en utilisant un éditeur de textes, ce qui permet de procéder à des modifications sans avoir à lancer un éditeur d'images.

Un tel fichier SVG est décrit à l'aide de balises XML servant chacune à la description d'une caractéristique d'une image.

Conformément à l'invention, à partir des fichiers source d'origine, il est procédé à une insertion de balises supplémentaires, par exemple au moyen d'un éditeur de textes conventionnel, ces balises servant à lancer l'exécution de programmes par l'étape de traitement 20 de manière à provoquer une modification des images les rendant compatibles avec un terminal sur lequel elle doit être affichée.



Ainsi, comme on le conçoit, les fichiers stockés dans la base d'images 12 sont constitués par des fichiers universels, c'est-à-dire des fichiers communs à l'ensemble des terminaux disponibles.

5 Les balises pouvant être utilisées pour l'enrichissement des fichiers d'image source peuvent être de différentes natures, en fonction des manipulations d'images à effectuer.

10 On notera, en particulier, que des balises de traduction, pour provoquer une traduction à la volée des éléments textuels en fonction de la langue utilisée par l'utilisateur peuvent être insérées dans les fichiers sources.

15 De même, des balises de modification de texture d'image, peuvent être prévues. De telles balises permettent une personnalisation de toute ou partie du contenu SVG en agissant sur des facteurs tels que les couleurs de remplissage, de fond, de tracé, de police, ..., les tailles de polices, les polices,...

20 Enfin, des balises de manipulation d'images peuvent être prévues. Par exemple, de telles balises peuvent être utilisées soit pour isoler une partie d'une image, par exemple pour afficher une portion d'un plan géographique sur un poste téléphonique cellulaire mobile ou sur un assistant personnel, pour lesquels la taille de l'écran ne permet pas un affichage d'un tel plan avec une échelle acceptable, soit pour procéder à l'exécution de fonctions géométriques, telles que des translations, des rotations, des homothétie, ... ou encore de fonctions d'agrandissement (positif ou négatif, sur toute ou partie de l'image).

25 À partir de ces images, le serveur, et en particulier l'étage de traitement 20, procède à l'élaboration d'images modifiées en procédant à un enrichissement du contenu initial des images par insertion dynamique de contenu.

30 En particulier, l'étage de traitement 20 récupère la valeur d'arguments véhiculés dans les requêtes transmises par les terminaux 14, 16, et 18, insère respectivement ces arguments dans les attributs des balises correspondantes, puis procède à l'exécution de programmes, en fonction des balises contenues dans les fichiers d'images et en fonction des arguments récupérés à partir des requêtes,

pour engendrer des codes d'instructions. Ces codes d'instructions sont alors insérés dans les fichiers en lieu et place des balises.

Il est alors créé, à ce stade, un fichier SVG spécifique à chaque terminal et répondant à l'ensemble des contraintes imposées pour l'affichage.

Un module de traitement 25, intégré à l'étage de traitement 20, est alors utilisé pour convertir le fichier SVG enrichi en des images susceptibles d'être lues par les terminaux 14, 16 et 18, par exemple des images se présentant sous les formats GIF, JPEG, PNG, TIFF,...

Les images ainsi élaborées sont alors stockées dans la mémoire cache 22 afin d'être disponible ultérieurement, pour être à nouveau téléchargées vers les utilisateurs en réponse à des requêtes similaires.

On va maintenant décrire la structure générale des fichiers chargés dans la base d'image 12. En particulier, on va décrire ci-après quelques exemples de balises pouvant être utilisées pour provoquer une modification des images par enrichissement de codes.

Dans ce qui suit, les balises servant à la description des images proprement dites sont constituées par des éléments de type classique à la portée d'un homme du métier et n'ont pas été reprises.

Comme indiqué précédemment, un fichier SVG est un document structuré conformément à une grammaire XML et doit donc commencer par une en-tête précisant la version de XML utilisée.

L'ensemble des éléments d'un fichier SVG est alors décrit à l'aide de balises. Les balises reconnues par le langage SVG doivent être définies dans un fichier DTD de définition de type de document ("Document type Definition"). Ainsi, les fichiers XML stockés dans la base d'images 12 doivent, avant tout, débiter par une référence vers ce document DTD.

Lors de l'élaboration de la base d'images 12, il est également procédé à une insertion des balises supplémentaires utilisées dans ce document.

En ce qui concerne les balises pouvant être intégrées au fichier source, les balises suivantes peuvent être utilisées :

- "Background Color" : cette balise permet d'ajouter une couleur d'arrière-plan à une image SVG. Cette balise s'appuie sur une valeur de paramètre "color" (couleur d'arrière-plan choisie), stockée dans une requête HTTP transmise par les terminaux. Ainsi, par exemple, le code utilisé pour insérer cette balise dans le fichier source sera de la forme :

```
<svg...>  
    < svg : background color/>  
10 </svg>
```

- "Black White" : cette balise définit une section d'image dans laquelle la gestion de la couleur va entrer en jeu. Elle permet d'appliquer un filtre noir et blanc sur une image vectorielle ou existante de type GIFF, JPG, ... Elle s'appuie sur la valeur d'un paramètre indiquant si un filtre noir et blanc doit être indiqué ou non de type booléen, stockée dans une requête HTTP. Ainsi, par exemple, le code utilisé pour insérer cette balise dans le fichier source sera de la forme :

```
20 <svg...>  
    <defs> ...</defs>  
        < svg : black white>  
        < /svg : black white>  
25 </svg>
```

- "Canevas" : cette balise repose sur un concept de "Surface virtuelle d'affichage". Elle permet de définir une surface servant de base de calcul pour générer la taille des objets contenus par cette surface. Elle s'appuie sur des valeurs de paramètres WIDTH et HEIGHT stockées dans une requête HTTP et sur des attributs WIDTH et HEIGHT pour calculer un ratio de re-dimensionnement. Ainsi, par exemple, le code utilisé pour insérer cette balise dans le fichier source sera de la forme :

```
<svg : canevas width= « 150 » height= « 100 »>  
<svg width= «150» height= « 100» viewBox= « 0 0 150 100»...>  
</svg>  
5 </svg :canevas>
```

10 - "Clipping" : cette balise permet de sélectionner une région dans une image SVG. Un rectangle de sélection permet de délimiter cette région. Cette balise s'appuie sur les valeurs de paramètres X et Y correspondant à des coordonnées X et Y d'un point situé à un coin supérieur gauche du rectangle, et d'un paramètre WIDTH et HEIGHT (largeur et hauteur du rectangle de sélection), stockés dans une requête HTTP. Ainsi, par exemple, le code utilisé pour insérer cette balise dans le fichier source sera de la forme :

```
15 <svg...<svg :clipping/>...>  
</svg...>
```

20 - "Encoder" : cette balise définit une section où l'utilisateur va paramétrer un transcodeur. Cette balise utilise l'adresse URL du transcodeur à utiliser et doit elle-même être intégrée à une autre balise. Ainsi, par exemple, le code utilisé pour insérer cette balise dans le fichier source sera de la forme :

```
25 <svg : encoder name= « svg » url= « http://xxxxxx.jsp »>  
</svg :encoder>
```

30 - Encoder\_Parameter : cette balise permet d'ajouter un paramètre qui sera transmis à la source à transcoder. Ainsi, dans le cas où un flot à transcoder serait issu de l'exécution d'un élément de programme JAVA, celui-ci pourra être paramétré à volonté. Par exemple, le code utilisé pour insérer cette balise dans le fichier source sera de la forme :

```

<svg>
  <svg : encoder unit url = « http://xxxx/transcoder/transcoder »>
    <svg : encoder name= « svg » url= « http://xxxxxx.jsp »>
      <svg : encoder parameter name= « width » value = « 100 »
5      <svg : encoder parameter name= « height » value = « 100 »
      <svg : encoder parameter name= « mime » value = « image/
                                          png »
    </svg :encoder>
  </svg :encoder unit>
10 </svg>

```

- Encoder-Unit : cette balise définit une section où l'utilisation d'un transcodeur va entrer en jeu. On pourra se référer à l'exemple ci-dessus pour obtenir une description d'une telle balise.

15 - Rescale-Hight : cette balise permet de calculer la hauteur d'un objet, contenu dans un canevas, à partir de sa hauteur initiale, qui correspond à un attribut de la balise. Ainsi, par exemple, le code utilisé pour insérer cette balise dans le fichier source sera de la

20 forme :

```

<svg...height= «< svg : rescale height value = « 100 »/ >
                                viewBox =«0 0 150 100 »...>
</svg>

```

25 - Rescale-Width : cette balise permet de calculer la largeur d'un objet, contenu dans un canevas à partir de sa largeur initiale, correspondant à un attribut de la balise. Ainsi, par exemple, le code utilisé pour insérer cette balise dans le fichier source sera de la

30 forme :

```

<svg...width= «< svg : rescale width value = « 150 »/ >...
                                viewBox =«0 0 150 100 »...>
</svg>

```

5       - "Rotation" : cette balise définit une section d'image dans laquelle une rotation de l'image va entrer en jeu. Elle s'appuie sur des valeurs de paramètres "Angle" (angle de rotation) X et Y (coordonnées d'un point de rotation), stockées dans une requête HTTP. Par exemple, le code utilisé pour insérer cette balise dans le fichier source sera de la forme :

```
10           <svg...>
              <defs> ...</defs>
              <svg :rotation>
              </svg :rotation>
              </svg...>
```

15       - "Translation" : cette balise définit une section ou une translation géométrique de l'image va entrer en jeu. Elle s'appuie sur des valeurs de paramètres TX et TY (translation selon les directions X et Y), véhiculées par une requête HTTP. Par exemple, le code utilisé

20       pour insérer cette balise dans le fichier source sera de la forme :

```
              <svg...>
              <defs> ...</defs>
              <svg :translation>
25            </svg :translation>
              </svg...>
```

30       - "Zoom" : cette balise permet de réaliser un agrandissement sur une région d'une image SVG en indiquant un niveau d'agrandissement. Cette balise s'appuie sur des valeurs de paramètres X et Y, qui correspondent à des coordonnées X et Y d'un point, dans l'image d'origine, à partir duquel le zoom va être réalisé, des valeurs de paramètre WIDTH et HEIGHT, qui correspondent à la largeur et à la hauteur de l'image de destination résultant du zoom, et d'un

paramètre zoom, qui sont stockées dans la requête HTTP issue des terminaux. Cette balise doit elle-même être contenue dans une balise « Canevas ». Par exemple, le code utilisé pour insérer cette balise dans le fichier source sera de la forme :

5

```
<svg :canevas width= « 800 » height= « 600 »>  
<svg... < svg:zoom />...>  
</svg>  
</svg :canevas>
```

10

- "Zoom\_Auto" : cette balise permet de zoomer sur une région d'une image SVG de deux façons :

15 - soit en indiquant un rectangle de sélection dans l'image d'origine, défini à partir d'un point de coordonnées X1, Y1, qui correspond à un coin supérieur gauche du rectangle, d'une largeur et d'une hauteur, ce rectangle représentant la région dans laquelle le zoom va être réalisé, et la largeur et la hauteur de l'image de destination représentant le résultat du zoom ;

20 - soit en indiquant un rectangle de sélection dans l'image d'origine, défini à partir de deux points de coordonnées X1, Y1 et X2, et Y2, ce rectangle représentant la région dans laquelle le zoom va être réalisé, d'un bord horizontal et d'un bord vertical optionnels, ajoutés à la taille du rectangle, la largeur et la hauteur de l'image de destination représentant le résultat du zoom.

25

Cette balise doit elle-même être contenue dans une balise « Canevas ». Par exemple, le code utilisé pour insérer cette balise dans le fichier source sera de la forme :

30

```
<svg :canevas width= « 800 » height= « 600 »>  
<svg... < svg:zoom_auto />...>  
</svg>  
</svg :canevas>
```

- « decode » : Cette balise permet de décoder le texte contenu dans son corps. On notera que cette balise n'utilise pas d'argument mais analyse le texte contenu dans son corps.

5           - « encode » : Cette balise permet de coder le texte contenu dans son corps. On notera que cette balise n'utilise pas d'argument mais analyse le texte contenu dans son corps.

10           - « translation\_unit » : Cette balise définit une section dans laquelle une traduction de texte va être effectuée. Elle utilise une constante « SecuredvalkyrieConstants.language\_request\_param » qui définit la langue courante sélectionnée par l'utilisateur. Pour une description de cette balise, on se référera à l'exemple de balise ci-dessous.

15

          « Translate » : Cette balise permet de traduire un texte contenu dans son corps de balise. Par exemple, le code de cette balise est le suivant :

20           <translation\_unit source= « mail/xsl/inbox »>  
          <b><translate text= « WELCOME »/></b>  
          ...  
          <i><translate text= « CONTINUE »/> ?</i>  
          <i>  
25           <translate text= « YES »/>  
          <translate text= « NO »/>  
          </i>  
          </translation\_unit>

30           Comme indiqué précédemment, ces balises, qui utilisent des attributs ou des informations véhiculées par les requêtes HTTP transmises à partir des terminaux 14, 16 et 18, provoquent la génération de codes SVG et le remplacement de ces balises par le code



ainsi élaboré. Bien entendu, d'autres balises peuvent être utilisées en fonction des contraintes à résoudre ou des buts à obtenir.

On notera que, en raison du format utilisé, les moyens de génération de codes peuvent être constitués par des éditeurs de textes, moins coûteux en moyens de programmation que des éditeurs d'images.

On va maintenant décrire en référence à la figure 2 les principales étapes d'un procédé d'élaboration et de transmission dynamique d'images conforme à l'invention.

Tout d'abord, au cours d'une première étape 26, il est procédé à l'élaboration des fichiers de la base d'images 12 à partir d'un fichier d'origine, par insertion des balises.

Dès réception, par le serveur, d'une requête (étape 28) tendant à la récupération d'une image, l'étage de traitement 20 demande à l'étage de gestion 24 de mémoire cache si une telle image a déjà été générée (étape 30).

Si tel est le cas, l'étage de gestion 24 extrait l'image de la mémoire cache 22 et la transmet à l'étage de traitement 20 (étape 32), lequel la retransmet directement au terminal ayant établi la requête (étape 34).

Si tel n'est pas le cas, c'est-à-dire si aucune image correspondant n'est stockée dans la mémoire cache 22, lors de l'étape 36 suivante, l'étage de traitement 20 récupère de la base d'images 22 une image et récupère les attributs ou les éléments d'informations véhiculés par la requête http transmise par le terminal 14, 16 ou 18. Comme indiqué précédemment, cette image se présente sous la forme d'un fichier SVG, décrit à l'aide de balises XML, et comportant en outre des balises supplémentaires la rendant universelle, c'est-à-dire convenant a priori à tout type de support, sous réserve d'une manipulation.

Lors de l'étape 38 suivante, l'étage de traitement récupère les programmes correspondants aux balises supplémentaires introduites dans le fichier d'image récupéré puis lance l'exécution de ces programmes en fonction des attributs transmis par la requête pour engendrer du code SVG supplémentaire. Il est alors procédé à une

insertion de ce code dans l'image, aux lieux et places des balises concernées.

5 À partir de ce code SVG, l'étage de traitement 20 met en œuvre le module de traitement 25 pour générer une image susceptible d'être directement affichée sur l'écran du terminal concerné, par exemple sous un format GIF, JPEG, TIFF, ....

10 Lors de l'étape 40 suivante, l'étage de traitement 20 transmet l'image ainsi élaborée au terminal et à l'étage de gestion 24 pour être mémorisée dans la mémoire cache 22. L'image pourra ainsi être récupérée ultérieurement, à la demande.

On notera que, dans le cas où le terminal 14 est constitué par un micro-ordinateur, lequel dispose d'un écran d'affichage compatible avec le format SVG, le module de traitement 25 n'est pas mis en œuvre le code SVG étant alors directement transmis vers le terminal.

15 De même, le traitement réalisé par ce module de traitement 25, peut, le cas échéant, directement être effectué au sein des terminaux, dans le cas où leur capacité le permet.

20 On notera enfin que les documents présents sous le format XML séparent généralement le fond et la forme de l'image et ne véhiculent dès lors aucune information d'affichage. Ainsi, de préférence, l'étage de traitement 20 incorpore des moyens de traitement de type XSLT ("Extensible Style Sheet Language Transformation"), de manière à appliquer une feuille de style aux images pour les compléter en insérant des informations d'affichage.

25 Par exemple, la feuille de style utilisée est une feuille de style en langage de programmation JAVA ou une feuille de style JSP ("Java Serveur Page"), de manière à introduire un mécanisme d'insertion d'informations d'affichage en amont, c'est-à-dire avant l'élaboration des images en format GIF, JPEG, PNG, TIFF, ..., affichables  
30 directement sur les écrans des terminaux 14, 16 et 18.

Enfin, de préférence, une seule feuille de style est prévue pour l'ensemble des terminaux. Cette feuille de style est manipulée dynamiquement par l'étage de traitement 20 en fonction des informations transmises à partir de chaque terminal distant. Cette

feuille de style est alors automatiquement adaptée aux contraintes de chaque terminal, ce qui permet d'améliorer la puissance du procédé et de diminuer son coût de mise en œuvre.

## REVENDICATIONS

1. Système d'élaboration dynamique d'images et de transmission de ces images vers des terminaux distants (14, 16, 18), comprenant un serveur comportant une base d'images (12) et des  
5 moyens (12, 20, 25) de traitement des images destinées aux terminaux adaptés pour convertir les images en formats respectifs compatibles avec les terminaux distants (14, 16, 18), caractérisé en ce que la base d'images est une base d'images vectorielles et en ce que les moyens de  
10 traitement comportent un ensemble de balises intégrées aux images vectorielles et adaptées pour provoquer une manipulation des images pour les rendre compatibles avec les terminaux distants et des moyens (25) pour générer des images lisibles par les terminaux à partir des images manipulées.

2. Système selon la revendication 1, caractérisé en ce que les  
15 images stockées dans la base d'images (12) sont des images structurées selon une grammaire XML.

3. Système selon la revendication 2, caractérisé en ce que les images sont structurées selon le format SVG.

4. Système selon l'une des revendications 2 et 3, caractérisé  
20 en ce que les moyens de traitement (20, 25) comportent en outre un ensemble de programmes de traitement d'images activés sélectivement par lesdites balises pour modifier en conséquence une image extraite de la base d'images (12).

5. Système selon la revendication 4, caractérisé en ce que les  
25 moyens de traitement comportent en outre des moyens d'élaboration d'une feuille de style adaptée pour insérer dans les images des informations de présentation.

6. Système selon la revendication 5, caractérisé en ce que les  
30 feuilles de style constituent des moyens de transformation de type XSLT.

7. Système selon l'une des revendications 5 et 6, caractérisé en ce que les moyens d'élaboration de la feuille de style comportent

des moyens pour élaborer la feuille de style en fonction d'informations transmises à partir des terminaux distants.

5 8. Système selon l'une quelconque des revendications 1 à 7, caractérisé en ce que les moyens (25) pour générer les images lisibles sont hébergées dans le serveur.

9. Système selon l'une quelconque des revendications 1 à 7, caractérisé en ce que les moyens pour générer les images lisibles sont hébergées dans les terminaux (14, 16, 18).

10 10. Système selon l'une quelconque des revendications 1 à 9, caractérisé en ce que le serveur comporte en outre une mémoire cache (22) pour le stockage des images élaborées par les moyens de traitement.

15 11. Système selon l'une quelconque des revendications 4 à 10, caractérisé en ce que les balises comportant chacune au moins un argument, lesdits programmes de traitement constituent des moyens pour provoquer une insertion de codes d'instruction dans l'image extraite de la base d'images (12) en fonction du ou de chaque argument.

20 12. Système selon la revendication 11, caractérisé en ce que les arguments des balises sont véhiculés par des requêtes transmises par les terminaux distants (14, 16, 18).

13. Procédé d'élaboration dynamique d'images et de transmission de ces images vers des terminaux distants, caractérisé en ce qu'il comporte les étapes consistant à :

- 25 - stocker des images vectorielles dans une base d'images (12) gérée par un serveur, les images comprenant chacune des balises adaptées pour provoquer une manipulation des images pour les rendre compatibles avec des terminaux distants (14, 16, 18) pour l'affichage des images ;
- 30 - modifier les images en fonction des balises insérées ; et
- générer des images lisibles par les terminaux à partir des images dans lesquelles sont insérer les balises.

14. Procédé selon la revendication 13, caractérisé en ce qu'au cours de l'étape de modification des images, pour chaque balise

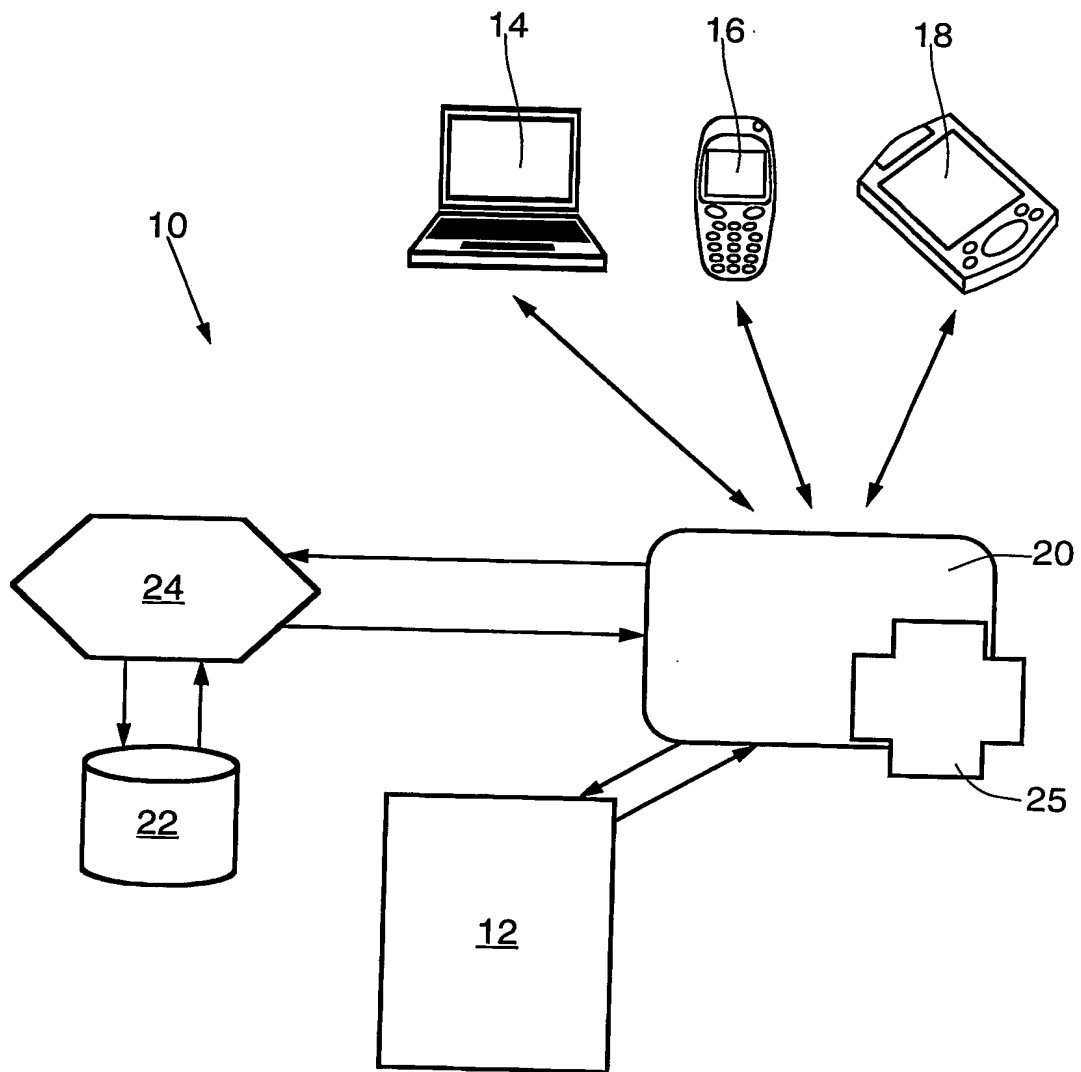
détectée dans les images, on active un programme de traitement d'image spécifique.

5 15. Procédé selon la revendication 14, caractérisé en ce qu'au cours de l'étape de modification des images, le programme de traitement provoque l'élaboration d'un code de programmation et l'enrichissement dynamique d'un fichier d'image par insertion dudit code en remplacement des balises.

10 16. Procédé selon l'une quelconque des revendications 13 à 15, caractérisé en ce que les images lisibles par les terminaux (14, 16, 18) sont élaborés par lesdits terminaux.

17. Procédé selon l'une des revendications 13 à 15, caractérisé en ce que les images lisibles par les terminaux sont élaborés par le serveur.

1/2

FIG.1

2/2

FIG.2